Toward Virtual Machine Packing Optimization Based on Genetic Algorithm

Hidemoto Nakada, Takahiro Hirofuchi, Hirotaka Ogawa, and Satoshi Itoh

National Institute of Advanced Industrial Science and Technology,
Tukuba, Ibaraki 3058568, Japan
{hide-nakada,t.hirofuchi,h-ogawa,satoshi.itoh}@aist.go.jp

Abstract. To enable efficient resource provisioning in HaaS (Hardware as a Service) cloud systems, virtual machine packing, which migrate virtual machines to minimize running real node, is essential. The virtual machine packing problem is a multi-objective optimization problem with several parameters and weights on parameters change dynamically subject to cloud provider preference. We propose to employ Genetic Algorithm (GA) method, that is one of the meta-heuristics. We implemented a prototype Virtual Machine packing optimization mechanism on Grivon, which is a virtual cluster management system we have been developing. The preliminary evaluation implied the GA method is promising for the problem.

1 Introduction

Cloud systems can be categorized into several types, including Hardware as a Service (HaaS) type, which provides (virtualized) hardware directly to the consumer. Representative example of this type is Amazon EC2. HaaS cloud systems often map several virtual machines onto a single real node for resource utilization. Virtual machines live-migration allows cloud provider to dynamically change the mapping for better system utilization. In this document, we call the optimization of the mapping the Virtual machine Packing Problem. The optimization problem have several parameters, includes,

- **Avoid SLA (Service level agreement) violation**
  To comply SLA is the first priority.
- **Reduce number of real nodes in use**
  Reducing number of real nodes used will result reducing energy consumption.
- **Reduce Virtual Machine migrations**
  Virtual Machine (VM) migration is not free. It consumes network bandwidth and will affect the performance of the migrated VM.

The parameters and weighting on each parameter could change dynamically depending on cloud provider preference and run-time environments. Especially on multi-sited cloud system, where a cloud is hosted on several independent datacenters, VM migration cost varies depending on environment, which makes optimization complicate.

S. Omatu et al. (Eds.): IWANN 2009, Part II, LNCS 5518, pp. 651–654, 2009.

To solve this kind of problems, powerful and flexible optimization techniques are required. We propose to employ Genetic Algorithm (GA) method. GA is one of the meta-heuristics, which are known to be robust for dynamically changing objective functions.

2 Virtual Cluster Management System: Grivon

We implemented a prototype GA based optimization mechanism on Virtual Cluster Management System Grivon [1], [2].

Grivon provides 'virtual clusters', where not only computer resources but also network and storage resources are also virtualized. Virtual Cluster users make reservation via Web interface, specifying number of nodes, packages to be deployed and configured, and date and time to be used. On the reserved time, Grivon automatically set up virtualized computers, network and storage, provision OSs and packages with Rocks[3], the cluster provisioning tool developed in UCSD, and provides the 'virtual cluster' to users. Grivon Uses Xen [4] for computer virtualization, iSCSI and LVM for storage virtualization, tagged VLAN and VPN for network virtualization.

Grivon Provides RESTful interface, i.e. using simple XML over HTTP protocol, any module can control Grivon from outside. The reservation Web interface is implemented on top of the RESTful interface with JavaScript codes embedded in the HTML page.

Virtual Machine Packing with GA
We have implemented a Virtual Machine packing module for Grivon. The module monitors Grivon and periodically tries to optimize the packing with GA. Monitoring requests and migration orders are placed via the RESTful interface. The module takes current VM locations as input and outputs a series of migration orders.

3 Design of the GA Method

Chromosome Modeling. The chromosome is modeled as a sequence of real node representation, which is represented as a list of VMs hosted on the node. Figure 1 demonstrates a modeling for a specific configuration of virtual machines. The stars in the figure denote VMs. For example, Real node No.1 hosts 2 VMs of Virtual Cluster 2 and 1 VM of Virtual Cluster 3. As a result, the node is represented as `[2 2 3]`.

Crossover. We employed One-Point Crossover, i.e., randomly select real node, and cut two chromosomes there, and exchange one of the pieces. The difficulty is that, simple exchange will cause most children to have corrupted chromosome, as seen in Figure 2. The corrupted chromosome in the figure has too many VMs for Virtual Cluster 2, while having to few VMs for Virtual Cluster 3.

To avoid this, we introduced repair mechanism, which randomly selects too many VM and remove it, then randomly add VM that was appears too few.

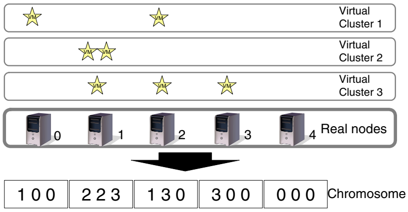

Fig. 1. Chromosome Modeling

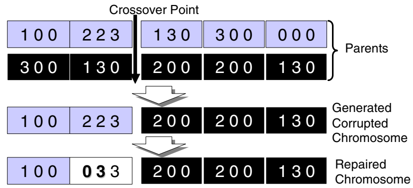

Fig. 2. Crossover with repair

Mutation. As mutation, we randomly pick two locations in chromosome and exchange the number stored in the locations. In case the two stored number is the same, meaning no change will happen, we discard the locations and retry to ensure all the mutations will cause real change in the individual represented by the chromosome.

Generation Alternation. For generation alternation mechanism, we implemented MGG (Minimal Generation Gap) [5] method and regular normalized weighted roulette method. We compared them and employed former because of better result.

Initial Population Generation. As described in the previous section, the module gets the current VM locations as input. We generate initial population from the input with repeatedly applying mutation on it.

4 Implementation and Preliminary Experiment

We implemented the optimization module based on GA with Java. The optimization module interacts with Grivon using the RESTful interface provided by Grivon. We also implemented GUI in JavaFX language to visualize migration

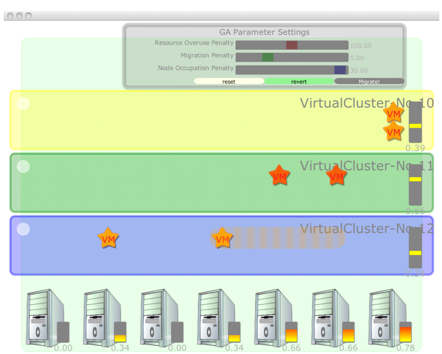

Fig. 3. GUI for GA based VM packing

behavior and to provide tuning tabs on optimization parameters to the users. Figure 3 shows the GUI in use for experiments. Here, as the objective function, we used a weighted-summation with; 1) penalty for violating SLA, 2) penalty for using extra nodes, and 3) penalty for migrating nodes. Administrator can change the weight for each parameter with GUI panal to get favorable results.

As the result of the preliminary experiment, we confirmed that the optimization time with GA it less than few seconds, which is short enough compared with the time spent for virtual machine migration which takes more than ten seconds to finish.

5 Conclusion

We proposed GA based Virtual Machine packing optimization for better utilization of resources in HaaS-type cloud system. While this study is still in very early stage, the prototype implementation showed that the GA method was flexible and fast enough for VM packing problems and promising.

As the future work we will 1) perform controlled experiments on the GA method, 2) apply multi-objective GA method for multi-objective optimization, and 3) apply this method to multi-sited virtual cluster system.

References

1. Nakada, H., Yokoi, T., Ebara, T., Tanimura, Y., Ogawa, H., Sekiguchi, S.: The design and implementation of a virtual cluster management system. In: Proc. of 1st IEEE/IFIP International Workshop on End-to-end Virtualization and Grid Management, pp. 61–71 (2007)
2. Hirofuchi, T., Yokoi, T., Ebara, T., Tanimura, Y., Ogawa, H., da, H.N.: Multi-site virtual cluster: A user-oriented, distributed deployment and management mechanism for grid computing environments. In: Proceedings of the Fourth IEEE/IFIP International Workshop on End-to-end Virtualization and Grid Management, pp. 203–216. Multicon Verlag (2008)
3. Papadopoulos, P.M., Katz, M.J., Bruno, G.: Npaci rocks: Tools and techniques for easily deploying manageable linux clusters. In: Cluster 2001: IEEE International Conference on Cluster Computing (2001)
4. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauery, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: SOSP 2003 (2003)
5. Sato, H., Yamamura, M., Kobayashi, S.: Minimal generation gap model for gas considering both exploration and exploitation. In: Proc. 4th Int'l. Conf. on Fuzzy Logic, Neural Nets and Soft Computing, pp. 494–497 (1996)